PRACTICAL-5

Aim:- Run rag locally using free embedding, phis index

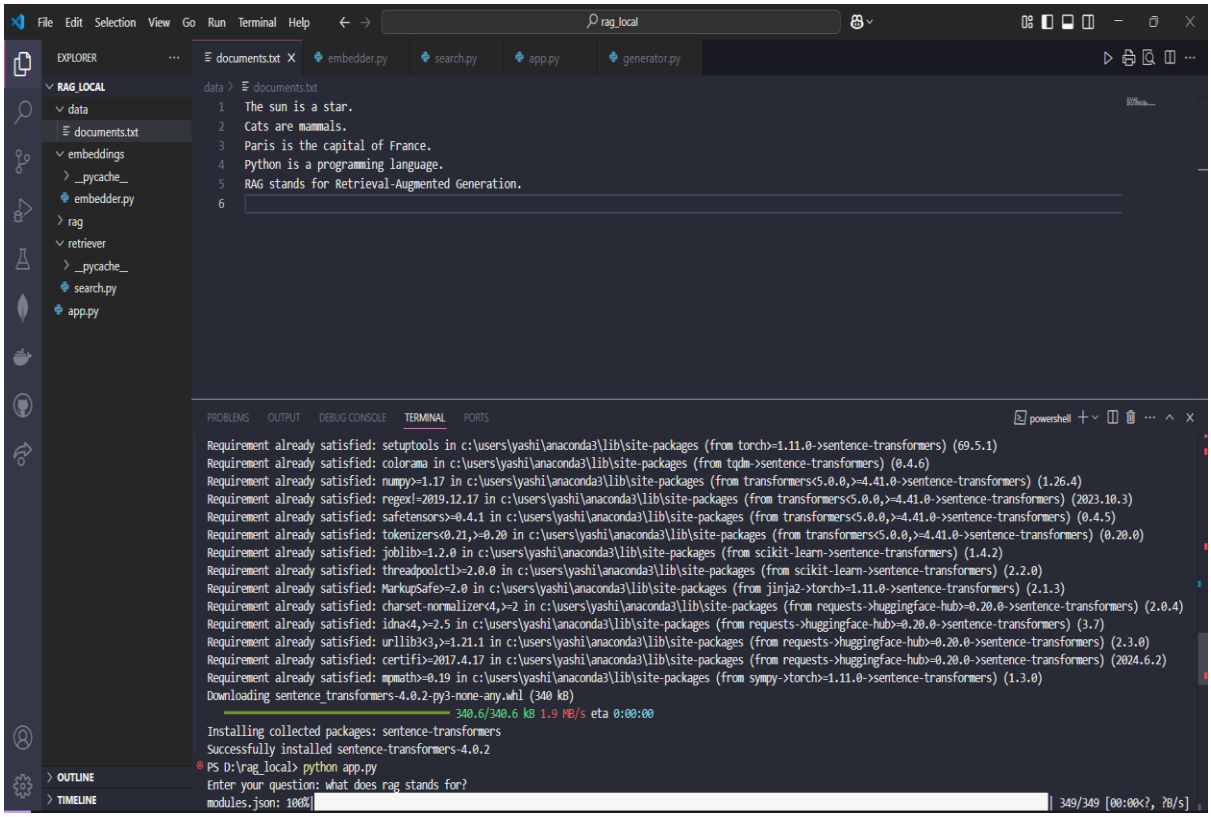
RAG (Retrieval-Augmented Generation) is a technique that improves LLM responses by retrieving relevant documents from a knowledge base. Instead of relying only on the model's memory, RAG fetches real data and passes it to the LLM for a grounded answer.
You **embed** documents into vector representations using an **embedding model** (like OpenAI's or Hugging Face's free ones). Then you **index** these vectors with **FAISS** (Facebook AI Similarity Search), which lets you quickly find the closest documents to any query.

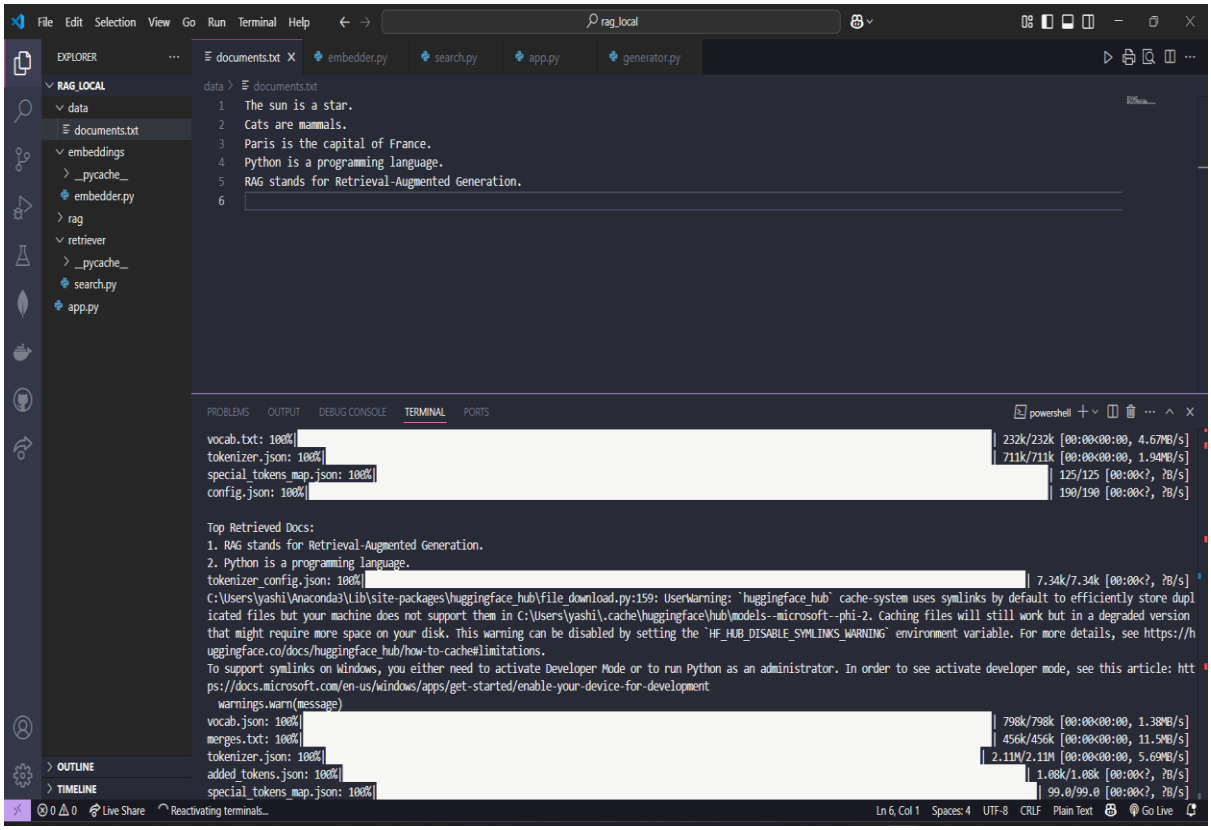
This process makes your chatbot smarter, more accurate, and capable of referencing *fresh knowledge* without retraining!

STEPS:-

1. Install Python (if not installed) and create a virtual environment.
2. Install libraries: faiss-cpu, sentence-transformers, langchain, and transformers.
3. Download or prepare your knowledge base (PDFs, text files, etc.).
4. Load and split your documents into smaller chunks (e.g., 500 characters each) using LangChain's text splitters.
5. Use a free **sentence-transformers** model (like all-MiniLM-L6-v2) to embed the document chunks into vectors.

6. Create a **FAISS** index and add all the document vectors into it.
7. Save the FAISS index locally for later use.
8. When a user sends a query, embed the query using the same embedding model.
9. Search the FAISS index for the most similar document chunks to the query.
10. Pass the retrieved documents + the user's question into an LLM (like HuggingFace flan-t5, or an API like OpenAI) to generate a final answer.
11. (Optional) Build a simple API or Streamlit frontend for local chat interaction.


File Edit Selection View Go Run Terminal Help
rag_local
EXPLORER
documents.txt
embedder.py
search.py
app.py
generator.py
RAG_LOCAL
data
documents.txt
embeddings
pycache
embedder.py
rag
retriever
pycache
search.py
app.py
1 The sun is a star.
2 Cats are mammals.
3 Paris is the capital of France.
4 Python is a programming language.
5 RAG stands for Retrieval-Augmented Generation.
6
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
Requirement already satisfied: setuptools in c:\users\yashi\anaconda3\lib\site-packages (from torch>=1.11.0->sentence-transformers) (69.5.1)
Requirement already satisfied: colorama in c:\users\yashi\anaconda3\lib\site-packages (from tqdm->sentence-transformers) (0.4.6)
Requirement already satisfied: numpy>=1.17 in c:\users\yashi\anaconda3\lib\site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers) (1.26.4)
Requirement already satisfied: regex!=2019.12.17 in c:\users\yashi\anaconda3\lib\site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers) (2023.10.3)
Requirement already satisfied: safetensors>=0.4.1 in c:\users\yashi\anaconda3\lib\site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers) (0.4.5)
Requirement already satisfied: tokenizers<0.21,>=0.20 in c:\users\yashi\anaconda3\lib\site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers) (0.20.0)
Requirement already satisfied: joblib>=1.2.0 in c:\users\yashi\anaconda3\lib\site-packages (from scikit-learn->sentence-transformers) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\yashi\anaconda3\lib\site-packages (from scikit-learn->sentence-transformers) (2.2.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\yashi\anaconda3\lib\site-packages (from jinja2->torch>=1.11.0->sentence-transformers) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\yashi\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.20.0->sentence-transformers) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\yashi\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.20.0->sentence-transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\yashi\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.20.0->sentence-transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\yashi\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.20.0->sentence-transformers) (2024.6.2)
Requirement already satisfied: mpmath>=0.19 in c:\users\yashi\anaconda3\lib\site-packages (from sympy->torch>=1.11.0->sentence-transformers) (1.3.0)
Downloading sentence_transformers-4.0.2-py3-none-any.whl (340 kB)
340.6/340.6 kB 1.9 MB/s eta 0:00:00
Installing collected packages: sentence-transformers
Successfully installed sentence-transformers-4.0.2
PS D:\rag_local> python app.py
Enter your question: what does rag stands for?
modules.json: 100%| 349/349 [00:00<?, ?B/s]
OUTLINE
TIMELINE


File Edit Selection View Go Run Terminal Help
rag_local
EXPLORER
documents.txt
embedder.py
search.py
app.py
generator.py
RAG_LOCAL
data
documents.txt
embeddings
pycache
embedder.py
rag
retriever
pycache
search.py
app.py
1 The sun is a star.
2 Cats are mammals.
3 Paris is the capital of France.
4 Python is a programming language.
5 RAG stands for Retrieval-Augmented Generation.
6
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
vocab.txt: 100%| 232k/232k [00:00<00:00, 4.67MB/s]
tokenizer.json: 100%| 711k/711k [00:00<00:00, 1.94MB/s]
special_tokens_map.json: 100%| 125/125 [00:00<?, ?B/s]
config.json: 100%| 190/190 [00:00<?, ?B/s]
Top Retrieved Docs:
1. RAG stands for Retrieval-Augmented Generation.
2. Python is a programming language.
tokenizer_config.json: 100%| 7.34k/7.34k [00:00<?, ?B/s]
C:\Users\yashi\Anaconda3\Lib\site-packages\huggingface_hub\file_download.py:159: UserWarning: `huggingface_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\yashi\.cache\huggingface\hub\models--microsoft--phi-2. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environment variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations.
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to see activate developer mode, see this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development
warnings.warn(message)
vocab.json: 100%| 798k/798k [00:00<00:00, 1.38MB/s]
merges.txt: 100%| 456k/456k [00:00<00:00, 11.5MB/s]
tokenizer.json: 100%| 2.11M/2.11M [00:00<00:00, 5.69MB/s]
added_tokens.json: 100%| 1.08k/1.08k [00:00<?, ?B/s]
special_tokens_map.json: 100%| 99.0/99.0 [00:00<?, ?B/s]
OUTLINE
TIMELINE
0 0 Live Share Reactivating terminals...
Ln 6, Col 1 Spaces: 4 UTF-8 CRLF Plain Text Go Live